

**AMENDMENTS TO THE CLAIMS**

*Please enter the following amendments:*

1. (Previously Presented) A software processing method comprising:

a monitoring step, which monitors a status of use of a resource identified as used by a process for a processor;

a determining step, which determines the status of use of the resource based upon contention information obtained in the monitoring step; and

a substituting step, which substitutes an equivalent process for a process using the resource, based upon the results of the determining step; wherein

the equivalent process is for the processor, is equivalent to the process using the resource, and makes reduced use of the resource.

2. (Canceled)

3. (Original) The software processing method according to claim 1, wherein: the resource is a storing device for a process, and the monitoring step for the status of use monitors the status of use of the storing device.

4. (Canceled)

5. (Previously Presented) The software processing method according to claim 3, wherein: the monitoring step for status of use stores the preceding statuses of use of the storing device corresponding to a plurality of entries so that the contention information is generated based upon the stored status and current status of use.

6. (Canceled)

7. (Original) The software processing method according to claim 3, wherein: the monitoring step for status of use stores the time of use when the storing device is in use, and based upon whether or not the time of use is not less than a predetermined value, the contention information is generated.

8. (Canceled)

9. (Original) The software processing method according to claim 1, wherein: the resource comprises a storing device for a process and a bus that connects the processor to the storing device, and the monitoring step for status of use monitors the status of use of the bus.

10. (Canceled)

11. (Previously Presented) The software processing method according to claim 9, wherein: the monitoring step for status of use stores the preceding statuses of use of the bus corresponding to a plurality of entries so that the contention information is generated based upon the stored status and current status of use.

12. (Canceled)

13. (Original) The software processing method according to claim 9, wherein: the monitoring step for status of use stores the time of use when the bus is in use, and based upon whether or not the time of use is not less than a predetermined value, the contention information is generated.

14. (Canceled)

15. (Original) The software processing method according to claim 1, wherein: the resource is a second processor that executes a process in response to a processing request from the processor, and the monitoring step for status of use monitors the status of use of the second processor.

16. (Currently Amended) A software processing method comprising:

a monitoring step for status of use, which monitors the status of use of a second processor, the second processor performing processing in response to a processing request by a first processor; and

an altering step for software processes, which alters software processing processes executed by the first processor or the second processor in response to contention information, the contention information being obtained in the monitoring step for status of use, wherein

the first processor can access a plurality of memory banks of a memory for the first processor by using a same address, and the plurality of memory banks includes a first memory bank used for the first processor and a second memory bank, the first memory bank including a program providing the processing request to the second processor, and the second memory bank including a program executed by the first processor; and

the contention information is a signal that indicates memory bank switching from ~~one of the plurality of memory banks to the memory bank used for the first processor~~ the first memory bank to the second memory bank.

17. (Canceled)

18. (Previously Presented) The software processing

a monitoring step, which monitors a status of use of a resource identified as used by a process for a processor;

a storing step for storing contention information obtained in the monitoring step at a current time;

a determining step, which determines the status of use of the resource based upon contention information at a past time; and

a substituting step, which substitutes an equivalent process for a process using the resource, based upon the results of the determining step; wherein

the equivalent process is for the processor, is equivalent to the process using the resource, and makes reduced use of the resource.

19. (Previously Presented) The software processing method according to claim 1, wherein: the contention information is processing time from the issuance of the processing request for the resource until the completion of the process, and the determining process for the status of use is a process which compares the processing time to a preset value.

20. (Original) The software processing method according to claim 18, wherein: the contention information is processing time from the issuance of the processing request for the resource until the completion of the process, and the determining process for the status of use is a process which compares the processing time to a preset value.

21. (Previously Presented) The software processing method according to claim 1, wherein: the contention information is waiting time from the issuance of the processing request for the resource until the start of the process, and the determining process for the status of use is a process which compares the waiting time to a preset value.

22. (Original) The software processing method according to claim 18, wherein: the contention information is waiting time from the issuance of the processing request for the resource until the start of the process, and the determining process for the status of use is a process which compares the waiting time to a preset value.

23. (Previously Presented) The software processing method according to claim 1, wherein the determining process for status of use reexamines the determination for the status of use of the resource regularly or irregularly.

24. (Original) The software processing method according to claim 18, wherein the determining process for status of use reexamines the determination for the status of use of the resource regularly or irregularly.

25. (Original) The software processing method according to claim 23, wherein the determining process for status of use reexamines the determination for the status of use of the resource by using random numbers.

26. (Original) The software processing method according to claim 24, wherein the determining process for status of use reexamines the determination for the status of use of the resource by using random numbers.

27. (Currently Amended) The software processing method according to claim 18,  
~~wherein~~ further comprising:

a process identifying step for identifying whether or not a process uses the resource from the software; and

an appearance portion identifying step for identifying the portions of appearance of the processes identified by the process identifying step, in the case when processes to be extracted by the process identifying ~~process~~ step are extracted from a plurality of portions of the software, ~~the compiler further adds to the software an identifying process for identifying the portions of appearance of the processes identified by the process identifying process~~

wherein the storing step stores the contention information for each of the portions of appearance so that the determining step carries out the determination by using the contention information stored for each of the portions of appearance.

28 - 31 (Canceled)

32. (Previously Presented) The software processing method according to claim 16, further comprising the steps of:

compiling a software:

identifying a process which uses the second processor in the software; and

mapping a process of using the first processor to the memory bank used for the first processor, and a process of using the second processor to a memory bank used for the second processor, wherein

the signal that indicates memory bank switching indicates switching to the memory bank used for the first processor.

33. (Previously Presented) The software processing method according to claim 32, further comprising the step of:

locking an operation of memory bank switching so that a memory bank switching cannot take place in case the first processor or the second processor is performing processing.

34. (Previously Presented) The software processing method according to claim 33, farther comprising the step of:

unlocking the locked operation when the first processor or the second processor has finished the processing so that the signal that indicates memory bank switching can be accepted.



35. (Currently Amended) A software processing system comprising:

a first processor;

a second processor for performing processing in response to a processing request by the first processor;

a use status monitoring device for monitoring a use status of the second processor;

a plurality of memory banks of a memory for the first processor that are accessible by the first processor by using a same address, the plurality of memory banks including a first memory bank used for the first processor and a second memory bank, the first memory bank including a program providing the processing request to the second processor, and the second memory bank including a program executed by the first processor; and

a bank switching device for altering software processing processes performed by the first processor or the second processor in response to contention information, the contention information being obtained by the use status monitoring device, wherein

the contention information is a signal that indicates memory bank switching from ~~one of the plurality of memory banks to the memory bank used for the first processor~~ the first memory bank to the second memory bank.

36. (Previously Presented) The software processing system according to claim 35, further comprising:

a compiler, which compiles a software;

a process identifier for identifying a process which uses the second processor in the software,

a mapper for mapping a process of using the first processor to the memory bank used for the first processor, and a process of using the second processor to a memory bank used for the second processor, wherein

the signal that indicates memory bank switching indicates switching to the memory bank used for the first processor or to the memory bank used for the first processor.

37. (New) A compiler for a processor monitoring a status of use of a resource, the compiler comprising:

a process identifier that identifies a process using the resource in an input program;

a determiner that determines a status of use of the resource; and

a substituter that substitutes an equivalent process which is equivalent to the process identified by the process identifier and does not use the resource for the process identified by the process identifier based upon a determination by the determiner.

38. (New) The compiler according to claim 37, wherein the determiner determines a status of use of the resource for each appearance portion of the process identified by the process identifier.

39. (New) The compiler according to claim 37, wherein the resource is a bus connecting the processor with a memory.

40. (New) The compiler according to claim 37, wherein the resource is a memory connected to the processor.

41. (New) The compiler according to claim 37, wherein the resource is a second processor that performs processing in response to a request from the processor.